**A cartoon of a snake wearing a biohazard suit

Description automatically generated**

**Protect Your Python:**

**SBOMs and Open Source Scanning**

**20 minutes total**

**1. Introduction (2 minutes)**

* **Opening Remarks**: Welcome audience and introduce the topic.
* **Agenda Overview**: Briefly outline the presentation structure.

**Opening Remarks:** Thank you for joining us today; we're excited to discuss an essential aspect of modern software development and cybersecurity: Software Bill of Materials (SBOMs) and open source scanning tools.

**Agenda Overview:** In this presentation, we will cover:

1. Free and Open Source Software (FOSS) risks
2. The Software Bill of Materials (SBOM)
3. Open source scanning tools.
4. Using scanning tools and SBOMs to enhance security and compliance.
5. Example/Walkthrough
6. Conclusion/Q&A

**2. Open Source Risks (2m)**

Vulnerabilities can occur in any programming language. Python is a great example because it is so widely used, especially in open source machine learning tools. With people of all skill levels using it, the risks are more pronounced and interesting to look at.

Examples of recently reported vulnerabilities in open source Python tools:

<https://thehackernews.com/2024/02/new-malicious-pypi-packages-caught.html>

* New Malicious PyPI Packages Caught Using Covert Side-Loading Tactics

<https://www.sonatype.com/blog/top-8-malicious-attacks-recently-found-on-pypi>

* RAT (remote access trojan) mutants
* PyTorch namespace (dependency) confusion attack
* GTA 5 multihack site linked to the httops package
* How stolen information stealers are fueling an underground market

**3. Understanding SBOMs (5 minutes)**

* **Definition of SBOM**: Explain Software Bill of Materials (SBOM).
* **Importance of SBOM**: Highlight the significance in software development and cybersecurity.
* **Components of SBOM**: List typical components (e.g., software packages, licenses, versions).

**Benefits**

* **Transparency**: Enhancing software transparency.
* **Risk Management**: Identifying and managing vulnerabilities.
* **Compliance**: Ensuring legal and regulatory compliance.
* **Efficiency**: Streamlining software maintenance and updates.

**4. Open Source Scanning Tools (4 minutes)**

* **Definition and Purpose**: Explain what open source scanning tools are and their role.
* **Key Features**: Highlight key features (e.g., license detection, vulnerability scanning).

**Popular Tools**

* **Tool 1: OWASP Dependency-Check**: Overview and key features.
* **Tool 2: Snyk**: Overview and key features.

**4. Integrating SBOMs with Open Source Scanning Tools (2 minutes)**

* **Workflow Integration**: How to integrate SBOMs with scanning tools in CI/CD pipelines.
* **Automation Benefits**: Advantages of automation in vulnerability management and compliance.

**5. Example/Walkthrough (5 minutes)**

Active community and consistent maintenance efforts help in promptly addressing vulnerabilities, but the risk is still non-zero. To mitigate these risks, it’s essential to use the latest versions, apply security patches, and perform regular vulnerability scans. The following example demonstrates how to perform a scan.

**Scenario:** You are developing a machine learning model using TensorFlow, a widely-used open source library.

**Steps:**

1. **Initial Scan:**
   * Run an open source scanning tool (e.g., Snyk, OWASP Dependency-Check) to identify known vulnerabilities in the TensorFlow library and its dependencies.
2. **Identify Vulnerabilities:**
   * The scan detects a critical vulnerability in a specific version of TensorFlow that could allow remote code execution.
3. **Update Dependencies:**
   * Based on the scan results, you update TensorFlow to a version where the vulnerability is patched.
4. **Continuous Monitoring:**
   * Implement continuous scanning in your CI/CD pipeline to monitor for new vulnerabilities in dependencies.
5. **Compliance and Licensing:**
   * The scan also checks for licensing issues, ensuring all components comply with your project's license policy.

**Benefits:**

* **Proactive Risk Management:**
  + By identifying and addressing vulnerabilities early, you reduce the risk of exploitation in production environments.
* **Maintaining Security Posture:**
  + Continuous monitoring helps maintain a strong security posture by staying updated with the latest security patches.
* **Ensuring Compliance:**
  + Automatic checks for license compliance prevent legal issues related to using open source components.

Using open source scanning tools as part of your development process ensures that you are aware of and can address security risks associated with the libraries and dependencies you use.

* **Outcomes and Lessons Learned**: Highlight results and key takeaways.

**6. Conclusion and Q&A (3 minutes)**

* **Summary of Key Points**: Recap the importance of SBOMs and open source scanning tools.
* **Q&A Session**: Open the floor for questions and discussions.